

Провел тесты libguestfs в различных режимах работы: uml, qemu-kvm, qemu. На основании полученных данных по потреблению памяти и времени работы лучшие результаты показывает режим работы, в котором в качестве backend-а используется UML.

Запуск UML выполняется в 2 раза быстрее по сравнению с qemu-kvm и в 4-5 раз быстрее по сравнению с qemu. Считывание большого файла (порядка 100Мб) UML осуществляет примерно наравных с KVM-QEMU (UML иногда немного медленнее), при использовании только QEMU считывание происходит гораздо медленнее. Считывании большого количества файлов малого размера (10000 файлов по ~1Кб) UML осуществляет быстрее всех, далее идет KVM\_QEMU и в конце QEMU.

По потреблению памяти UML снова показывает лучшие результаты: UML ~250Мб, KVM\_QEMU ~350Мб, QEMU ~500Мб

Насколько изменится производительность разработанного решения при использовании виртуальной машины UML?

Сразу скажу, что производительность, конечно, увеличится. UML, или иными словами User-Mode Linux, представляет собой такой вариант ядра Linux, который позволяет запускать операционную систему Linux в качестве обычного пользовательского приложения. UML, по своей сути, использует механизм паравиртуализации, т.е. запускаемая нами гостевая операционная система Linux знает, что работает не на «голом» железе как обычный Linux, а внутри другой операционной системы Linux, возможности которой использует для своей работы. Собственно, это является одной из главных причин, почему UML в своей работе показывает более хорошую производительность по сравнению с виртуальной машиной QEMU, которая в своей работе использует полную виртуализацию (позволяет запускать немодифицированные гостевые операционные системы), что естественно влечет более высокие накладные расходы.

Собственно, тестирование производительности работы библиотеки в операционной системе Linux показало, что при использовании UML скорость выполнения всех ключевых операций увеличивается. Так, например, время запуска UML сокращается в 2 раза, по сравнению с QEMU, а также существенно повышается скорость выполнения операций чтения и записи.

К сожалению, UML можно использовать только в операционной системе Linux, поэтому моя дальнейшая работа связана с портированием UML для работы в операционной системе Windows.

Время запуска libguestfs при использовании в качестве backend-а:

- uml: 1.4 с
- qemu\_kvm: 3.5 с
- qemu: 8 с

Монтирование образа диска (~1Гб):

- uml: 1.6 с
- qemu\_kvm: 0.7-0.8 с
- qemu: 6-7 с

Время считывания файла размером 100Мб:

- uml: 1.5 с
- qemu\_kvm: 0.4-0.5 с
- qemu: 3-5 с

Время считывания 10000 файлов размером 1Кб:

- uml: 16-19 с
- qemu\_kvm: 23-24 с
- qemu: 23-25 с

Какой объем оперативной памяти и дискового пространства потребляет разработанное решение?

Размер полученной библиотеки составляет 3 МБ, размер используемого дистрибутива QEMU равен 22 МБ. Количество оперативной памяти, потребляемой во время работы библиотеки зависит, главным образом, от параметров запуска виртуальной машины QEMU, в частности, от того, сколько оперативной памяти предоставляется запускаемому внутри виртуальной машины программному обеспечению (в нашем случае, это операционная система Linux с работающим внутри неё демоном). В общем случае, размер потребляемой оперативной памяти складывается из следующих факторов:

- [1] Размер библиотеки libguestfs (3 МБ)
- [2] Размер дистрибутива QEMU (22 МБ)
- [3] Размер оперативной памяти для ПО внутри QEMU (128 МБ)
- [4] Размер внутренних структур, используемых в процессе работы QEMU (40 МБ)

Так, например, при предоставлении Linux 128 МБ QEMU в сумме потребляет 190 МБ, т.е. накладные расходы QEMU дополнительно составляют около 40 МБ.

Также, при использовании разделяемой памяти для передачи файлов количество потребляемой оперативной памяти увеличивается соответственно размеру используемой общей памяти.

Размер libguestfs и образа UML:

- libguestfs: ~5 МБ
- UML: 37.5 МБ
- QEMU: 6 МБ

Qemu: 22 + 30 (dep) MB

Libguestfs: 3 + 2.5 MB

Отсутствует сравнение скорости выполнения операций чтения и записи файлов при помощи полученной библиотеки и скорости выполнения операций чтения и записи файлов при помощи существующих Windows-драйверов.

Сравнение с существующими драйверами не было произведено потому, что по размеру потребляемой памяти и скорости выполнения операций чтения и записи мы, очевидно, проиграем, поскольку запуск виртуальной машины с работающей внутри неё операционной системой предполагает достаточно высокие накладные расходы.

На самом деле было произведено сравнение с одним из существующих наиболее успешных Windows-драйверов Extfsd. Как видно на слайде, все операции выполняются быстрее, запись нескольких файлов медленнее, видимо из-за несовершенной реализации операции записи, что характерно для таких драйверов (потому как главным образом они предназначены для операции чтения).